A Gaussian Process Guided Particle Filter for Tracking 3D Human Pose in Video

Suman Sedai, Mohammed Bennamoun, Member, IEEE, and Du Q. Huynh, Member, IEEE

Abstract

In this paper, we propose a hybrid method that combines Gaussian Process learning, a particle filter, and annealing to track the 3D pose of a human subject in video sequences. Our approach, which we refer to as “Annealed Gaussian Process Guided Particle Filter” (AGP-PF), comprises two steps. In the training step, we use a supervised learning method to train a Gaussian Process regressor that takes the silhouette descriptor as an input and produces multiple output poses modelled by a mixture of Gaussian distributions. In the tracking step, the output pose distributions from the Gaussian Process regression are combined with the annealed particle filter to track the 3D pose in each frame of the video sequence. Our experiments show that the proposed method does not require initialization and does not lose tracking of the pose. We compare our approach with a standard annealed particle filter using the HumanEva-I dataset and with other state of the art approaches using the HumanEva-II dataset. The evaluation results show that our approach can successfully track the 3D human pose over long video sequences and give more accurate pose tracking results than the annealed particle filter.

Index Terms

3D human pose tracking, Gaussian Process regression, particle filter, hybrid method.

I. INTRODUCTION AND MOTIVATION

Image and video-based human pose estimation and tracking is a popular research area due to its large number of applications including surveillance, security and human computer interaction. For example, in video-based smart surveillance systems, 3D poses can be used to infer the action of a subject in a scene and detect abnormal behaviors. It can provide an advanced human computer interface for gaming and virtual reality applications. Also, 3D poses of a person computed over a number of video frames can be useful for biometric applications to recognize a person.
These applications require simple video cameras or still images as input and, as a result, provide a low cost solution in contrast to marker-based systems.

Human pose estimation and tracking systems can be broadly classified into three different approaches: discriminative, generative and hybrid methods.

In generative methods, the output pose is estimated by searching the solution space for a pose that best explains the observed image features [1], [2]. In this approach, a generative model is constructed which measures how close the hypothesized pose is to the observed image features. A hypothesized pose that is most consistent with the observed image features is chosen as the output pose. Particle filter [2], [3] is a generative tracking method to estimate the pose in each frame using an estimate of the previous frame and the motion information. Given an image of a human subject, there are multiple 3D human poses associated with the image. Such pose ambiguities can be resolved using images from multiple camera views [4].

In discriminative methods, a regression function that maps image features to the pose parameters is obtained using supervised learning [5], [6]. Although discriminative methods can quickly estimate the 3D pose, they can produce incorrect predictions for new inputs if the system is trained using small datasets. Moreover, the relationship between image features and the human pose is often multimodal. For example, when the human silhouette is used as an image feature, one silhouette can be associated with more than one pose, resulting in ambiguities. In such cases, multiple discriminative models are needed to build one-to-many relationships between image features and poses [5], [7].

Discriminative methods are powerful for specific tasks such as human pose estimation as they are only based on the mapping from the input to the desired output. The generative methods, on the other hand, are flexible because they provide room for using partial knowledge of the solution space and exploit the human body model to explore the solution space. Due to their different ways of predicting the final output, the two methods are considered to complement each other. Thus hybrid generative and discriminative methods have shown to have the potential to improve pose estimation performance. As a result, they have gained more attention recently.

In hybrid methods, such as the ones presented in Refs. [8], [9], a discriminative mapping function is used to generate a pose hypothesis space. The pose is then estimated by searching the hypothesis space using a generative method. The method of Ref. [8] is only based on pose estimation from a single image and is therefore unable to track the pose from a video sequence. Moreover, these methods assume that the pose hypothesis space generated from the discriminative model is always correct and hence fail to handle the case when the discriminative model predicts incorrect poses.

In this paper, we propose a hybrid discriminative and generative method to track the 3D human pose from both single and multiple cameras. For the discriminative model, we use a mixture of Gaussian Process (GP) regression
models, which are obtained by training GP models in different regions of the pose space. The GP regression has the advantage of being able to give a probabilistic estimate of the 3D human pose. It provides an effective way of incorporating more confident discriminative predictions into the tracking process while discarding the uncertain ones. To the best of our knowledge, this is the first hybrid method that takes into account the predictive uncertainty of the discriminative model and combines it with a generative model to improve pose estimation. We treat the probabilistic output from the GP regression as one component of the hypothesis space. In the tracking step, we combine this hypothesis space with the hypothesis space obtained from the motion model and the search for the optimal pose is performed using an annealed particle filter.

A major contribution of this paper is the introduction of a novel method to combine Gaussian Process regression and annealed particle filter for 3D human pose tracking. Our method can probabilistically discard uncertain predictions from the regression model and hence only use predictions that are likely to be correct to track the 3D pose. Moreover, our method can resolve ambiguities when motion information is used along with the image cues from multiple views during pose tracking.

The organization of the paper is as follows. Section II presents the related work. Section III presents the details of the training of the discriminative model. Section IV presents the proposed AGP-PF method for 3D human pose tracking. Section V provides the experimental results and Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

In discriminative estimation, a direct mapping between image features and the pose parameters is learned using a training dataset. Examples of discriminative learning includes nearest neighbor based (example based) regression [6] and sparse regression [10], [11]. Often the relationship between the image features and the pose space is multimodal. The multimodal relationship is established by learning the one-to-many mapping functions that are defined in terms of a mixture of sparse regressors [5], [7] or Gaussian Process regression [12], [13]. This setting produces a multiplicity of the pose solutions that can either be ranked using a gating function [14] or be verified using an observation likelihood [8] or be disambiguated using temporal constancy [11]. For effective pose estimation using a discriminative approach, it is important that image features are compact and distinctive. Approaches such as [15], [16] use a metric learning method to suppress the feature components that are irrelevant to pose estimation. Recent research shows that feature selection and pose estimation can be carried out using regression trees [17] and random forest [18]. Other methods such as [19], [20] use dimensionality reduction to make the feature vector more distinctive for pose estimation. The fusion of multiple image cues based on regression has also been used to improve the pose estimation performance [21], [22]. In another approach [23], 2D trajectories of the human limbs in a video are used as features and the mapping between the trajectory features and the 3D poses is modelled using Gaussian Process regression. It has also been shown that pose estimation performance can be improved by taking
into account the dependencies between the output dimensions, e.g., using structural SVM [24] and Gaussian Process regression [25]. Dimensionality reduction can also be used to address the correlation between output dimensions, e.g., the training of mapping function is performed in a lower dimensional subspace of features and pose so as to make use of unlabelled data [26].

In generative inference, the pose that best explains the observed image features is determined from a population of hypothesized poses. A search algorithm is used to search the prior pose space for the pose that exhibits the maximum likelihood value for the observed image feature [1], [27]. Generative methods thus consist of three basic components: a state prior (pose hypotheses), an observation likelihood model (matching function), and a search algorithm.

The search algorithm can be local or global in nature. Most of the local search algorithms employ Newton’s optimization method [4], [28]. Global search methods are based on stochastic sampling techniques where the solution space is represented by a set of points sampled according to some hypotheses. Successful global search algorithms include annealing [27], Markov Chain Monte-Carlo [29], covariance scaled sampling [1], and Dynamic Programming [30]. In order to refine the pose tracking, the sampling based global search techniques have been combined with some local search techniques [31], [32]. In order to reduce the search space, the pose prior space based on a predefined set of activities, such as walking and jogging, has been used [33]. Pose priors based on models that constrain the human body pose to physically plausible configurations have also been used to reduce the search space [34].

In order to compute the likelihood of a pose hypothesis, image features, such as silhouettes, edges [1], [27], color [29], [30] and learned features [35], have been used. First, the human body model corresponding to the hypothesized pose is projected onto the image feature space to obtain a model feature (a.k.a template feature). The likelihood value is computed as a similarity measure between the observed image feature and the template feature. For 3D pose estimation, human body models based on cylinders [33], [36], superquadrics [37] and 3D surfaces [4], [9] are commonly used. For 2D pose estimation, a simple rectangular cardboard model has been used [38], [39].

To reduce the complexity of tracking in higher dimensions, prior models based on a lower-dimensional representation of the pose have been used. The commonly used models include linear models such as the Principal Component Analysis [33] and non-linear approximation such as the Mixture of Factor Analysers [40]. In another approach [41], the Restricted Boltzmann Machine is used to model human motion in a discrete latent space. A more commonly used non-linear dimensionality reduction technique for pose tracking is the Gaussian Process Latent Variable Model (GPLVM) [42], [43], which does not always provide a smooth latent space as required for tracking. To overcome this problem, a Gaussian Process Dynamical Model (GPDM) [44] which uses non-linear regression to model the motion dynamics in a latent space in a GPLVM framework can be employed. Tracking in
the latent space often has a lower computational complexity; however, it has also been argued that models based on dimensionality reduction have limited capacity [45].

In the hybrid methods, discriminative and generative approaches are combined together to exploit their complementary power to predict the pose more accurately. To combine these two methods, the observation likelihood obtained from a generative model is used to verify the pose hypotheses obtained from the discriminative mapping functions for pose estimation [8], [9]. In other work, e.g., [45], generative and discriminative models are iteratively trained using Expectation Maximization (EM). In each step of the EM, the predictions made by one model are used to train the other model. Recently, the work of Ref. [46] combines the discriminative and generative models by applying distance constraints to the predictions made by the discriminative model and verifying them using image likelihoods. None of these approaches utilizes the prediction uncertainty of the discriminative model to track the 3D human pose. In this paper, we propose a hybrid method that takes into account the prediction uncertainty of the discriminative model and effectively combines it with the generative model.

We use Gaussian Process regression as the discriminative model. The probabilistic output poses from the discriminative model are integrated with a particle filter and a subsequent annealing step for 3D human pose tracking. Consequently, the pose tracking performance is improved, since the search space is reduced to only include the correct hypotheses generated from the discriminative model combined with the hypotheses of the motion model. This is an extension of our previous work [47], where we used a Relevance Vector Machine as a discriminative model combined with a particle filter for 2D pose tracking. GP regression has been used in Ref. [48] to learn the dynamical model and the observation likelihood model for object tracking. However, their method does not involve a mapping from the feature space to the output space. Our method, on the other hand, uses GP regression to learn a discriminative model which gives a conditional distribution of the 3D human pose. In this paper, we use a generic form of the motion model and observation likelihood model for tracking. In another purely discriminative filtering-based approach [49], the unreliability of the observation is modelled using a probabilistic classifier to regulate the predictions from multiple regressors. Our method, on the other hand, models the prediction uncertainty by the variance of the prediction from the Gaussian Process regressor and hence does not require a separate classifier to model the unreliability. Furthermore, unlike [49], our method incorporates the image likelihoods obtained from the generative model to verify the pose hypotheses and to estimate the target state distribution.

III. TRAINING STEP

In the training step, we use supervised learning to construct a multimodal mapping between the shape descriptor space and the 3D pose space. Once the mapping model is trained, the multimodal 3D poses can be estimated using the trained model. A block diagram showing the training of our mixture of Gaussian Process regressors is shown in Fig. 1. Given the training images, we first extract the silhouette images using background subtraction [50]. Then,
we extract the silhouette descriptors using Discrete Cosine Transform (DCT). We divide the 3D pose space into \( K \) clusters and we train the mapping from the silhouette descriptor space to the 3D pose space of each cluster. The final output of the training stage is a mixture of Gaussian Process regression models.

A. Feature and Pose Representation

A review of many shape and appearance descriptors that are applicable to discriminative pose estimation is available in Ref. [51]. In this paper, we use the silhouette as an image feature and the Discrete Cosine Transform (DCT) of the silhouette as the shape descriptor. We use the DCT because it is simple to compute and yet more discriminative than other shape descriptors [51], [52]. It is shown in Ref. [52] that the DCT descriptor outperforms other shape descriptors such as Histogram of Shape Contexts and Lipschitz embeddings for human pose estimation.

The DCT descriptor, which belongs to the family of orthogonal moments, represents the silhouette image by the sum of two dimensional cosine functions of different frequencies characterized by the various coefficients. The DCT has been popularly used for image compression. First a silhouette window is cropped from the foreground image obtained from background subtraction. As shown in Fig. 2, the cropped image is scaled to the size of \( H \times W \) and the DCT descriptor for the image window is computed as
\[ M_{p,q} = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} f_p(x) g_q(y) I(x, y), \]  

for \( p = 0, \cdots, W - 1; q = 0, \cdots, H - 1, \)

where \( f_p(x) = \alpha_p \cos \{ p\pi(x+0.5)/W \} \); \( g_q(y) = \alpha_q \cos \{ q\pi(y+0.5)/H \} \); and \( \alpha_p = \sqrt{(1 + \min(p, 1))/W} \) and \( \alpha_q = \sqrt{(1 + \min(q, 1))/H} \). We take \( W = 64 \) and \( H = 128 \) pixels. This window size is the most commonly used size for human subjects that are in an upright pose. The DCT descriptor has a nice property that most of the rich information about the silhouette is encoded in just a few of its coefficients. We empirically found that setting the descriptor to a 64 dimensional vector (corresponding to 8 rows and 8 columns of the DCT matrix \( M \)) is sufficient to represent each silhouette. In this paper, we assume that the subject is upright in the image. Although the DCT descriptor is not rotation invariant, it does not affect the pose estimation so long as the human subject is in an upright pose in the image. It is possible to handle athletic motions such as handstands and cartwheels by training a separate discriminative model for such activities and using a classifier to select the most appropriate model for an input feature. Since the silhouette is centered and re-scaled to the standard size, the descriptor is invariant to translation and scale.

We represent each 3D pose as the relative orientations of the body parts in terms of Euler angles. We take the torso as the root segment and the orientation of the torso segment is measured with respect to a global coordinate system. The orientations of the upper arms are measured relative to the orientations of the torso. The orientations of the lower arms are measured relative to the upper arms. Relative orientations between the upper legs and the torso and between the lower and upper legs are defined in a similar manner.

### B. Mixture of Gaussian Process (MGP) Regressors

We use a supervised learning technique to estimate the 3D pose directly from the silhouette descriptor. We learn the piecewise mapping of Gaussian Process (GP) regressor [53] from the shape descriptor space \( x \in \mathbb{R}^m \) to the 3D pose space \( y \in \mathbb{R}^d \) using the training data samples \( T = \{ x^{(i)}, y^{(i)} \} \), for \( i = 1, \cdots, N \), where \( N \) is the number of training samples. With such a mixture of Gaussian Process (MGP) regression model [54], the GP regressors are trained for each region of the data space and a separate classifier is used to select the GP model that is appropriate for an input feature. First, the 3D pose space is partitioned into \( K \) clusters using the hierarchical k-means algorithm. The training set is then divided into \( K \) subsets, \( T_1, \cdots, T_K \), such that \( (x^{(i)}, y^{(i)}) \in T_k \) if \( y^{(i)} \) belongs to the \( k \)th cluster. We assume that the components of the output vector \( y \) are independent of each other so we train the separate GP regression model for each output component of \( y = [y_1, \cdots, y_d]^T \). Without loss of generality, we drop the subscript \( q \) in \( y_q \) (which represents the \( q \)th component of \( y \)) and present only a one-dimensional Gaussian
Process regression. In each cluster, the relationship between \( x^{(i)} \) and each component \( y^{(i)} \) of the training pose instance \( y^{(i)} \) (the superscript \( (i) \) represents the \( i \)th training instance) is modeled by

\[
y^{(i)} = f_k(x^{(i)}) + \epsilon^{(i)}_k ,
\]

(2)

where \( \epsilon^{(i)}_k \sim \mathcal{N}(0, \beta^{-1}_k) \) and the \( \beta_k \) is the hyper-parameter representing the precision of the noise. From the definition of a Gaussian Process [53], the joint distribution of the output variables is given by a Gaussian:

\[
p(y_k | X_k) = \mathcal{N}(0, C_k),
\]

(3)

where \( y_k = [y^{(i)}, \ldots, y^{(N_k)}]^T, X_k = [x^{(i)}, \ldots, x^{(N_k)}]^T \) for all \( y^{(i)} \in k \)th cluster, and \( C_k \) is a covariance matrix whose entries are given by \( C_k(i, j) = \kappa(x^{(i)}, x^{(j)}) + \beta^{-1}_k \delta_{ij} \). The covariance function \( \kappa(x^{(i)}, x^{(j)}) \) can be expressed as

\[
\kappa(x^{(i)}, x^{(j)}) = \theta_{k,1} \exp\left(-\frac{\theta_{k,2}}{2} \left\| x^{(i)} - x^{(j)} \right\|^2 \right) + \theta_{k,3},
\]

(4)

where the parameters \( \Phi_k = (\theta_{k,1}, \theta_{k,2}, \theta_{k,3}, \beta_k) \) are referred to as the hyper-parameters of the GP and \( \delta_{ij} \) is the Kronecker’s delta function. This covariance function combines the Radial Basis Function (RBF) and a bias term. The learning of the hyper-parameters is based on the evaluation of the likelihood function \( p(y_k | \Phi_k) \) and the maximization of the log likelihood using a gradient based optimization technique such as conjugate gradient. The log likelihood function for a Gaussian process can be evaluated as

\[
\ln p(y_k | \Phi_k) = -\frac{1}{2} \ln |C_k| - \frac{1}{2} y_k^T C_k^{-1} y_k - \frac{N_k}{2} \ln (2\pi).
\]

(5)

Once the hyper-parameters are trained, the next step is to predict the output pose component \( y_{k*} \) for the unseen test feature vector \( x_* \). This requires the evaluation of the predictive distribution \( p(y_{k*} | y_k, X_k) \). Let us define \( y_{k*} = [y^{(1)}, \ldots, y^{(N_k)}, y_{k*}]^T \), whose joint distribution is given by

\[
p(y_{k*}) = \mathcal{N}(0, C_{k*}),
\]

(6)

where \( C_{k*} \in \mathbb{R}^{(N_k+1)\times(N_k+1)} \) is a covariance matrix. That is,

\[
C_{k*} = \begin{pmatrix}
C_k & c_{k*} \\
(c_{k*})^T & c_{k*}
\end{pmatrix},
\]

(7)

where the vector \( c_{k*} \) has elements \( c_{k*}(i) = \kappa(x^{(i)}, x_*) \), for \( i = 1, \ldots, N_k \), and the scalar \( c_k = \kappa(x_*, x_*) + \beta^{-1}_k = \theta_{k,1} + \theta_{k,3} + \beta^{-1}_k \) from Eq. (4). The conditional distribution \( p(y_{k*} | y_k, X_k) \) is a Gaussian distribution with mean
and variance given by

\[ y_{k^*} = c_k^T \Sigma_k^{-1} y_k \]  
\[ \sigma_k^2(x^*) = c_k - (c_k^*)^T \Sigma_k^{-1} c_k^* \].

In this manner, we obtain the prediction for each component of the pose vector. Let \( y_{q,k^*} \) be the prediction for the \( q \)th component of the pose vector and \( \sigma^2_{q,k}(x^*) \) be the corresponding variance. The full pose vector is \( y_{k^*} = [y_{1,k^*}, \ldots, y_{d,k^*}]^T \) and the corresponding covariance matrix becomes \( \Sigma_k(x^*) = \text{diag} \left( \sigma^2_{1,k^*}, \ldots, \sigma^2_{d,k^*} \right) \).

The multimodal relationship between the silhouette descriptor \( x \) and the 3D pose vector \( y \) is thus represented by a mixture of \( K \) regressors:

\[ p(y|x^*) = \sum_{k=1}^{K} g_k(x^*) \mathcal{N}(y_{k^*}, \Sigma_k(x^*)), \]  

where \( g_k(x) \) is a \( K \)-class classifier which gives the probability that the \( k \)th GP regressor is selected to predict the given feature instance \( x^* \). We model the multi-class classifier \( g_k(x) \) as a multinomial logistic regressor (i.e., the softmax function):

\[ g_k(x) = \frac{\exp(-v_k^T x)}{\sum_{j=1}^{K} \exp(-v_j^T x)} \],

where \( v_k \) is an \( n \)-dimensional parameter vector. The parameter vectors \( v_1, \ldots, v_K \) are estimated from the training data \( \{x^{(i)}, l^{(i)}\}_{i=1}^{N} \), where \( l^{(i)} = (l_1^{(i)}, \ldots, l_K^{(i)}) \) and \( l_k^{(i)} \) denotes the probability that feature vector \( x^{(i)} \) belongs to the \( k \)th cluster. We set \( l_k^{(i)} = 1 \) if \( y^{(i)} \) belongs to the \( k \)th cluster, otherwise we set it to 0. The maximum likelihood estimation of the parameter vectors is then performed using the iteratively reweighted least squares method. We use a fast method based on bound optimization described in [55] to train these parameter vectors.

Instead of the multinomial logistic regressor, an alternative is to model the multi-class classifier \( g_k(x) \) as a function of the variance of the \( k \)th GP mode, i.e., by setting \( g_k(x) \propto 1/\text{trace} \left( \Sigma_k(x) \right) \). As the average variance of the prediction is lower when the test feature is closer to the training samples, a higher weight is given to the cluster which is closer to the test feature vector. In our empirical evaluation shown in Table III, we found that the pose estimation performance of the MGP regressor is improved when the multinomial logistic regressor is used in comparison to the case when the function of the variance is used.

1) Pose Space Clustering: The motivation behind clustering the pose space and learning discriminative model in each cluster is to model the depth ambiguities introduced by the silhouettes. For that purpose, we cluster the pose space into six partitions using hierarchical k-means. At the first level, the pose space is partitioned into four clusters representing poses which face forward, backward, left and right with respect to the camera by a careful initialization. At the second level, each cluster that represents the lateral pose (left or right) is further partitioned into two clusters. Figure 3 shows the representative poses in each cluster along with the silhouettes that give rise
Fig. 3. Representative pose in each cluster obtained using hierarchical k-means. Each pose is rendered as a 3D cylindrical model with red denoting left limbs and blue denoting right limbs. Silhouettes that give rise to ambiguous poses are also shown (Figure best viewed in color).

to ambiguous poses. Clusters C1 and C2 model the forward-backward ambiguity as the silhouette S1 could be generated from the poses that are both in C1 and C2. Similarly, clusters C3 and C4 model the ambiguity associated with the silhouette S2; clusters C5 and C6 model the ambiguity associated with the silhouette S3. Other approaches address the ambiguities by sampling from the multimodal posterior obtained from a mixture of regressors models [5], [7]. In another interesting approach [56], it is shown that the ambiguous poses can be distinguished by sampling from the posterior in a latent space that is shared by both the observation and the pose spaces. In their approach, the posterior in the latent space is obtained using the GPLVM model.

This approach of hard clustering could result in a poor prediction performance at the boundary of the clusters. However, since we used the Gaussian Process regression as the discriminative model, such poor predictions could often be detected as they tend to produce larger prediction variances. In Section IV-C, we discuss the adoption of a probabilistic method to discard poor predictions made by the MGP regression model during the tracking of the 3D pose. The method of Ref. [12] provides an interesting solution to the boundary problem in that a MGP regressor is trained on the subset of the data that is closest to the test feature. However, their method requires computing the GP hyper-parameters for each test image.

IV. POSE TRACKING

In the tracking step, our goal is to determine the 3D pose of a person in the video frames. The block diagram of our proposed tracking method is shown in Fig. 4. For each image frame in a video, the human pose is predicted using a Mixture of Gaussian Process (MGP) regressors (Section III). The output of the MGP regressors is a mixture of Gaussian distributions. The output distribution is taken as one component of the hypothesis space. The other component being the output pose distribution of the previous frame. The output pose is then computed by searching the combined pose hypothesis space using our proposed AGP-PF tracking method.
Fig. 4. A block diagram showing the testing of our proposed 3D pose Tracking System. The 3D human pose for each video frame is predicted from the mixture of GP regressions; The tracking incorporates the predicted pose, prediction uncertainty, edges and silhouette observations using the AGP-PF method.

Fig. 5. A 3D human body model in a neutral pose. Each body part of the model is approximated by a tapered cylinder. The X-axis is denoted by a large circled dot and is perpendicular to both the Y- and Z-axes.

Our method needs a human body model to compute the likelihood of each pose hypothesis. In this paper, we use a 3D cylindrical model of the human body. Each body part is represented by a tapered cylinder as shown in Fig. 5. The length of the body parts and the diameters of the cylinders are assumed to be fixed for a given person and is initialized at the beginning of the tracking.

We use human kinematic constraints to determine the degrees of freedom (DOF) of the pose. The torso has six degrees of freedom after incorporating the global translation and rotation. The head, upper arms and upper legs have three degrees of freedom each. The forearms have one degree of freedom each (they are only allowed to rotate about their Y axis) and the lower legs have two degrees of freedom (they are allowed to rotate about their X and Y axes). Hence the human body model shown in Fig. 5 has 27 degrees of freedom. We enforce the joint angle limit by restricting the variation of the angles to a kinematically possible range. The kinematically possible angles are computed from the training data.

In this paper, we use our proposed Gaussian Process guided particle filter for pose tracking. Below, we will first
give a review of a standard particle filter (for completeness) in Section IV-B followed by our proposed Gaussian Process guided particle filter for pose tracking in Section IV-C.

A. Likelihood distribution

Given an image observation denoted by \( r_t \) at time \( t \), the likelihood density \( p(r_t | y_t) \) measures how well a hypothesized pose vector \( y_t \) explains the image observation \( r_t \). In our method, the likelihood value is computed by matching the 3D human body model corresponding to the hypothesized pose with the observed image features. We use the silhouette and edge features of the image to compute the likelihood of each pose state vector. We first project the 3D cylindrical model corresponding to the hypothesized pose onto the image plane using the camera calibration matrix to obtain the hypothesized edge features and the hypothesized region features. The likelihood value is computed by matching the hypothesized features with the observed image features. We use the silhouette and edge features of the observed image to compute the likelihood based on two cost measures: silhouette cost and edge cost, as detailed below.

1) Silhouette Cost: The silhouette cost measures how well the region projected by the hypothesized model fits into the observed silhouette. Given a 3D pose hypothesis, we first generate a binary image \( H \) of the corresponding 3D human body model such that \( H(x, y) = 1 \) if the pixel corresponds to the hypothesized foreground and \( H(x, y) = 0 \) otherwise. An example of a hypothesized foreground is shown in Fig. 6 (d). Let \( Z \) be the observed silhouette image as shown in Fig. 6 (b). The part of the silhouette \( Z \) that is not explained by the model region \( H \) is given by \( R_1 = Z \cap \bar{H} \), where \( \cap \) denotes the pixel-wise “and” operator and \( \bar{H} \) denotes the inverted image of \( H \). Similarly,
the part of the model region that is not explained by the silhouette is given by \( R_2 = H \cap \bar{Z} \). Since our objective is to minimize these unexplained silhouette and model regions, the cost can be expressed as

\[
C_{\text{sil}} = 0.5 \left( \frac{\text{Area}(R_1)}{\text{Area}(Z)} + \frac{\text{Area}(R_2)}{\text{Area}(H)} \right),
\]

(12)

where \( \text{Area}(I) \) gives the number of non-zero pixels in the binary image \( I \). When the hypothesized model fits the observed silhouette exactly then both \( \text{Area}(R_1) = \text{Area}(R_2) = 0 \) and hence \( C_{\text{sil}} \) will give the lowest cost of 0. When the two regions have zero overlap then no part of the silhouette region is explained by the model (i.e., \( Z \cap \bar{H} = Z \)) and no part of the model region is explained by the silhouette (i.e., \( H \cap \bar{Z} = H \)). In this case, \( C_{\text{sil}} \) will give the highest cost value of 1. This measure of the silhouette cost is similar to that of [57], [58].

2) Edge Cost: The edge cost measures how well the boundary line corresponding to the hypothesized model fits with the observed edge image. Given an input image, we detect edges by thresholding the gradient image to obtain a binary edge map [2]. We segment the foreground edges corresponding to the human subject by masking the binary edge image with the silhouette image. We then construct a Gaussian distance map \( E_1 \) of the segmented edge image to determine the edge probability of a given pixel. The Gaussian distance map, which gives the proximity of a pixel to the edge, can be obtained by convolving the binary edge image with a Gaussian kernel and rescaling the pixel values between 0 and 1 [2]. In the next step, we generate a set of hypothesized edge points \( E_2 \) by projecting the visible boundary of the 3D cylindrical model corresponding to a pose hypothesis to the edge image and sparsely sampling the points along the boundary. The points that are hidden due to self occlusion are discarded using the depth information of the body model. The edge cost is then obtained by computing the mean square error (MSE) of the edge probability values:

\[
C_{\text{edge}} = \frac{1}{|E_2|} \sum_{p \in E_2} (1 - E_1(p))^2.
\]

(13)

Similar to \( C_{\text{sil}} \), the value of \( C_{\text{edge}} \) falls between 0 and 1. Assuming equal influence of edge and silhouette features on tracking, the final likelihood for a given hypothesized pose is approximated by

\[
p(r_t|y_t) \approx \exp(-(C_{\text{sil}} + C_{\text{edge}})).
\]

(14)

For the case when images from more than one camera view are available, the silhouette and the edge costs are computed using images from each camera and the costs are averaged to obtain the final cost of a pose hypothesis in Eq. (14).
B. Particle Filter

The particle filter is a Monte Carlo approximation to the sequential Bayesian estimation, which propagates the posterior probability of the first order Markov process from time $t-1$ to $t$ through the following equation:

$$p(y_t|R_t) = c p(r_t|y_t) \int_{y_{t-1}} p(y_t|y_{t-1}) p(y_{t-1}|R_{t-1}),$$  \hspace{1cm} (15)

where $c$ is a constant, $y_t$ is the 3D pose state at time $t$; $r_t$ is the image observation; $R_t = [r_1, \ldots, r_t]$ comprises all observations observed sequentially up to time $t$; $p(y_t|y_{t-1})$ is a distribution that describes the motion model; and $p(r_t|y_t)$ is the observation likelihood distribution. The multi-dimensional integral of Eq. (15) can only be evaluated for the simple case where the posterior distribution of the state variable is Gaussian. When the state variable corresponds to the human pose, the posterior distribution is non-Gaussian and methods like the Kalman filters generally fail [59]. Particle filters are therefore often used to approximate Eq. (15) using a set of weighted samples $S_t = \{y^{(i)}_t, \pi^{(i)}_t\}_{i=1}^n$ where each $y^{(i)}_t$ is a particle, $\pi^{(i)}_t$ is the corresponding particle weight, which is normalized to ensure $\sum_i \pi^{(i)}_t = 1$ and $n$ denotes the number of particles. The particle filter does not make any explicit assumption about the form of the posterior and hence is applicable to systems where the posterior distribution of the state variable is non-Gaussian. In order to estimate the pose using the particle filter, one must design two models: a dynamical model (a.k.a. motion model), namely $p(y^{(i)}_t|y^{(i)}_{t-1})$ which describes the movement of the human subject from one frame to another in the 3D space; and the observation likelihood model $p(r_t|y_t)$ which gives the probability that observation $r_t$ can be generated by a pose sample $y_t$ as described in Section IV-A. At each time step $t$, given the particle set $S_{t-1}$, a basic sequential importance resampling updates the particles in three steps [60]. First, sample $n$ particles from the discrete distribution denoted by $S_{t-1}$ with replacement. In the second step, each sampled particle is modified by a motion model. In the third step, the normalized importance weight is computed from the observation likelihood. The new particle-weight set at time $t$ is obtained as $S_t$. The particle-weight set $S_t$ represents the posterior distribution of the state and the output can be computed by taking the expected value of the posterior distribution represented by the particle-weight set.

A simple particle filter does not work accurately in higher dimensions because a large number of particles is required to populate such a higher dimensional state space. Using a small number of particles might lead towards an entrapment of the particles around local maxima. The occurrence of local maxima in the state space is common in human pose tracking because there are many ways the model can partially fit the observed image. Variants of the particle filter, such as annealed particle filter [27] that iteratively pushes the particles towards the high probability regions of the state space, have been developed. Moreover, the computed importance weights might not always be correct, mainly because of the noisy/ambiguous observations and an incorrect generative model. This leads to
Fig. 7. Graphical model for AGP-FP method that includes the conditional $p(y_t|y_{t-1}, x_t)$ and the observation likelihood $p(r_t|y_t)$. We assume that $p(y_t|y_{t-1}, x_t)$ can be factored into the discriminative model $p(y_t|x_t)$ and the motion model $p(y_t|y_{t-1})$ according to Eq. (18). 

frequent mistracking. Our Annealed Gaussian Process guided particle filter described in the following subsection aims at tackling this problem.

C. Annealed Gaussian Process Guided Particle Filter

In Annealed Gaussian Process Guided Particle Filter (AGP-PF), not only the motion information is used, but the discriminative distribution obtained from the supervised learning in Section III is also used to track the pose from one frame to another. The graphical model for the AGP-PF is shown in Fig. 7. Let $y_t$ denote a hidden state that represents the 3D pose at time $t$ and $x_t$ be the image observation at time $t$ that is specifically used to generate the discriminative distribution $p(y_t|x_t)$. Let $r_t$ be the image observation at time $t$ that is used to compute the likelihood distribution $p(r_t|y_t)$. Let $X_t = [x_1, \ldots, x_t]$ and $R_t = [r_1, \ldots, r_t]$ be all observations which have been sequentially observed until time $t$. Then the posterior density of the state after a new observation $x_t$ and $r_t$ is given by a recursive Bayesian equation

$$p(y_t|R_t, X_t) = c \cdot p(r_t|y_t)p(y_t|R_{t-1}, X_t), \quad (16)$$

where $c = 1/p(r_t|R_{t-1}, X_t)$ is a constant term relative to $y_t$ and $p(r_t|y_t) = p(r_t|y_t, X_t)$ follows from the conditional independence according to the graphical model in Fig. 7. Since the integral of the target distribution $p(y_t|R_t, X_t)$ over the entire space of $y_t$ should be one, the target distribution can be calculated by first computing Eq. 16 without considering the constant $c$ and then normalizing it. Therefore, the constant $c$ does not influence the estimation of the target distribution.

The conditional independence assumption holds when the features $r$ and $x$ have different properties and they do not depend on each other. In our case, $x$ is taken as the Discrete Cosine Transform of the silhouette and hence it is a much coarser description of the shape. On the other hand, $r$ represents the edges of the body segments which corresponds to a richer appearance representation of the body parts. As the value of the DCT descriptor of a silhouette gives no knowledge about the value of the edges feature, these two features can be considered to be independent. The prior distribution at time $t$ can be written as
\[
p(y_t|R_{t-1}, X_t) = \int p(y_t|y_{t-1}, x_t)p(y_{t-1}|R_{t-1}, X_{t-1})dy_{t-1}.
\] (17)

We assume that the conditional distribution \( p(y_t|y_{t-1}, x_t) \) can be expressed as a mixture of simpler conditionals [61], i.e.,

\[
p(y_t|y_{t-1}, x_t) = (1 - \alpha(x_t))p(y_t|y_{t-1}) + \alpha(x_t)p(y_t|x_t),
\] (18)

where \( p(y_t|x_t) \) is a discriminative distribution expressed in terms of a mixture of Gaussians obtained from the trained Gaussian Process models (see Eq. (10)); \( \alpha(x_t) \in [0, 1] \) is a mixing coefficient that denotes the contribution of the discriminative distribution towards the prior at \( t \). We discard any contribution of the discriminative distribution \( p(y_t|x_t) \) when it has a large variance \( \sigma^2(x_t) \). This is achieved by relating the mixing coefficient \( \alpha(x_t) \) to the variance \( \sigma^2(x_t) \) using the following equation:

\[
\alpha(x_t) = \exp\left(-\frac{\sigma^2(x_t)}{\lambda}\right),
\] (19)

where \( \sigma^2(x_t) = \text{trace} [\Sigma_k(x_t)] / d \) is the average variance of the most likely GP model selected from Eq. (10).

The most likely GP model is the one with the highest gating probability \( g_k(x_t) \). Figure 9 shows the variations of \( \alpha(x_t) \) w.r.t \( \sigma^2(x_t) \) given in Eq. (19) for different values of \( \lambda \). The value of the control parameter \( \lambda \) is determined empirically as discussed in Section V-C1. From Eq. (19), it means that the prediction from the GP model that has a large variance (i.e., low certainty) is less likely to be correct and thus fewer samples should be drawn from it. From Eqs. (17) and (18), the prior distribution can be expressed as

\[
p(y_t|R_{t-1}, X_t) = (1 - \alpha(x_t))p(y_t|R_{t-1}, X_{t-1}) + \alpha(x_t)p(y_t|x_t),
\] (20)

where

\[
p(y_t|R_{t-1}, X_{t-1}) = \int p(y_t|y_{t-1})p(y_{t-1}|R_{t-1}, X_{t-1})dy_{t-1},
\] (21)

which is the first term of the prior distribution is obtained by applying the motion model to the posterior distribution at \( t - 1 \). The second term of the prior is the discriminative density \( p(y_t|x_t) \) given by Eq. (10). We approximate the posterior at time \( t - 1 \) by the particle-weight set \( S_{t-1} = \{y_{t-1}^{(i)}, \pi_{t-1}^{(i)}\}_{i=1}^{n} \) and assume a statistically stationary motion model, i.e., \( p(y_t|y_{t-1}) \sim \mathcal{N}(y_{t-1}, \Omega) \), where \( \Omega \) is a diagonal covariance matrix. A method to determine the value
of \( \Omega \) is discussed in Section V-C1. We can write the first term of the prior distribution as

\[
p(y_t|R_{t-1}, X_{t-1}) = \sum_{i} \pi_{t-1}^{(i)} N(y_{t-1}^{(i)}, \Omega).
\]

The second term of the prior distribution is the discriminative distribution obtained from supervised learning. The final prior of Eq. (20) can then be expressed as

\[
p(y_t|R_{t-1}, X_{t}) = (1 - \alpha(x_t)) \sum_{i=1}^{n} \pi_{t-1}^{(i)} N(y_{t-1}^{(i)}, \Omega) + \alpha(x_t)p(y_t|x_t), \tag{22}
\]

where the discriminative distribution \( p(y_t|x_t) \) is given by Eq. (10). We use the silhouette and edge based likelihood to model the likelihood distribution \( p(r_t|y_t) \) as described in Section IV-A. The importance weight for each particle can be computed as a normalized likelihood value, i.e.,

\[
\pi_{t}^{(i)} \propto \left( p(r_t|y_{t}^{(i)}) \right)^{A_l}, \tag{23}
\]

where \( p(r_t|y_{t}^{(i)}) \) is given by Eq. (14), \( A_l \) is the inverse annealing temperature for the \( l^{th} \) annealing layer, and \( \pi_{t}^{(i)} \) is normalized so that \( \sum_{i=1}^{n} \pi_{t}^{(i)} = 1 \). Simulated annealing works on the principle that a region of high probability lies in the vicinity of the particles which have higher weights [2]. At the first annealing layer, particles are allowed to float in high energy state. This means that all the particles have similar probabilities associated with them resulting in a smooth and non-peaky likelihood function. Then the system is gradually allowed to cool down at the successive annealing layer. By doing so, peaks in the likelihood function are introduced slowly. Consequently, more particles are concentrated around the high probability region at the end of the annealing layer. We automatically find the value of the annealing temperature \( A_l \) so that the particle survival rate at each layer is around 50\% (following the method of [2]):

\[
\zeta(A_l) = \frac{1}{n} \left( \sum_{i=1}^{n} \left( \pi_{t}^{(i)} \right)^2 \right)^{-1}, \tag{24}
\]

where \( A_l \) is estimated by minimizing \( |\zeta(A_l) - 0.5| \). This process of decreasing the particles survival rate has the same effect as the cooling of a mechanical system.

The steps of our proposed method are given in Table I. Let \( S_{t,l} \) be the particle weight set output at the end of an \( l^{th} \) annealing layer of frame \( t \). For \( t = 0 \), the initial set of particles \( S_0 \) is obtained by sampling from the discriminative distribution \( p(y_0|x_0) \) and equal importance weights are assigned to them. For \( t > 0 \), the input to the first annealing layer is the posterior distribution of the previous frame, i.e., \( S_{t,0} = S_{t-1} \). Then, at each annealing layer, the prior distribution is constructed from the discriminative distribution \( p(y_t|x_t) \), the mixing coefficient \( \alpha(x_t) \) and the discrete distribution \( S_{t,l-1} \) using Eq. (22). For layer \( l = 1 \), we compute the value of \( \alpha(x_t) \) according to Eq. (19). For \( l > 1 \), we force \( \alpha(x_t) = 0 \) to prevent any sampling from \( p(y_t|x_t) \) because samples from \( p(y_t|x_t) \) are
The particles are computed according to Eq. (23). At the end of each layer, the covariance matrix already taken at $A^{l−1}$ is shrunk by a factor of $0.1$. A pose is obtained by sampling from the first term of the prior distribution given in Eq. (22). In this way, all the samples in concatenating the global translation vector at the beginning of the pose vector. The global translation vectors are different from the camera view of the training images. Let $\tau$ be the torso orientation of a sample $y_t$ obtained from the discriminative distribution. To align the pose sample $y_t$ to the camera view of the test image, we compute $\tau ← \tau - \Delta$ where $\Delta$ is the camera direction angle difference between the training and test views. The angular difference $\Delta$ is determined from the camera calibration matrices of the two camera views. Also, each sample obtained from $p(y_t|x_t)$ is a pose vector in terms of Euler angles. We convert the pose vector to a motion vector by concatenating the global translation vector at the beginning of the pose vector. The global translation vectors are obtained by sampling from the first term of the prior distribution given in Eq. (22). In this way, all the samples in the particle set are obtained as a motion vector.

It can be seen that the annealed particle filter (APF) is a special case of an AGP-PF when $\alpha = 0$. When $\alpha = 1$, the method does not use the motion model; instead, it performs pose detection by sampling from the discriminative distribution alone and validating the samples using the importance weights computed from the likelihood distribution.

Hence, the method of Ref. [8] can be seen as a special case of an AGP-PF when $\alpha = 1$. Our method, on the other
hand, adaptively chooses the value of $\alpha$ between 0 and 1 as the mixing coefficient $\alpha(x_t)$ in Eq. (20) is inversely related to the prediction uncertainty of the discriminative model. Hence the pose predictions that are likely to be correct are retained to guide the tracking process. This provides a stable tracking since, at each time step, even if the motion model produces wrong samples, the samples obtained from the discriminative distribution are used to compute the posterior. Also, if the predictions from the discriminative model are uncertain, less emphasis is given to them. In such cases, the tracking is more driven by the results of the previous frame.

V. Experiments

A. Dataset Description

We trained and evaluated our proposed 3D human pose tracking method using the HumanEva-I data set [62]. We used video frames and corresponding 3D poses of the three subjects from the Walking and Jogging sequences of the dataset to train and evaluate our approach. The dataset was originally partitioned into training, validation and testing sets. However, as the ground truth of the testing set was not provided, we used the validation set (which provides ground truth) as our testing set and the original training set as our training set. Table II shows the number of images in the training and testing set for each activity. For each image, the corresponding ground truth 3D motion is given by the 3D pose and the global translation vector. The 3D pose is given by the relative orientations of the 10 body parts in terms of Euler angles. We also used the Walking data of subject S2 from the HumanEva-II dataset to compare our method with other ones.

B. Training

For all the images in the training and testing sets, we extracted the silhouette images using the background subtraction method described in Ref. [50]. We computed 64-dimensional DCT shape descriptor vectors from the silhouettes following the process described in Section III-A. We then trained the mixture of Gaussian Process regressors which map the DCT descriptor to the 3D poses using the supervised learning approach described in Section III-B. We set the number of clusters to $K = 6$ so as to allow sufficient partitioning of the pose space and to model the potential ambiguities associated with the silhouettes as described in Section III-B1. We trained a Gaussian Process regressor for each cluster. The final output of the discriminative model is the mixture of Gaussians given by Eq. (10).

C. Tracking and Evaluation

The discriminative model is combined with the motion model to track the pose using our proposed AGP-PF algorithm described in Section IV. We sampled 300 particles at each iteration of AGP-PF. The final output of the
AGP-PF is a 33-dimensional motion vector whose components are the global translation vector and the relative Euler angles of the ten body parts. Given the length of the body parts, we converted the motion vector to the absolute 3D joint locations using forward kinematic. We computed the 3D error between the estimated 3D joint locations $\hat{y}$ and the ground truth 3D joint locations $\tilde{y}$ as follows [58]:

$$E(y, \hat{y}) = \frac{1}{J} \sum_{i=1}^{J} \|m_i(\tilde{y}) - m_i(\hat{y})\|,$$

where $m_i(y) \in \mathbb{R}^3$ denotes the three dimensional coordinates of the $i^{th}$ joint location from the pose vector $y \in \mathbb{R}^d$; $\|\cdot\|$ denotes the Euclidean distance; and $J$ is the number of joint locations. The formula measures the 3D error in mm between the two pose vectors. The mean 3D error of the $T$ test images is computed by $\bar{E} = \frac{1}{T} \sum_{i=1}^{T} E(\tilde{y}^{(i)}, \hat{y}^{(i)})$.

We investigated three cases for pose tracking. The first is an AGP-PF where the value of the mixing coefficient $\alpha$ is set to be inversely proportional to the variance of the most likely Gaussian component of the discriminative distribution given in Eq. (19). In this case, $\alpha$ takes values between 0 and 1; When the variance of the discriminative distribution is lower, $\alpha$ is set to a higher value. Consequently, more samples are taken from the discriminative distribution during tracking. Conversely, a higher variance of the discriminative distribution sets $\alpha$ to a lower value. As a result, less samples are taken from the discriminative distribution. Since a higher variance of the discriminative distribution implies an uncertain prediction and vice-versa, the discriminative distribution which are more certain are automatically selected for tracking. Figure 8 shows an example of the inverse relationship between the variance of the discriminative distribution and the mixing coefficient $\alpha$ for each image in the video. It can be seen that the value of $\alpha$ varies according to the image frames.

In the second case, we set $\alpha = 1$, which suppresses the sequential sampling nature of AGP-PF. The search is only performed in the pose space predicted by GP regression. The search discards the output pose of the preceding frame. In this case, the motion model is only used to predict the global translation vectors. We refer to this case as the Annealed Gaussian Process (AGP) method.

In the third case, we set $\alpha = 0$ so only the motion model is used to predict the pose. In this case, the discriminative
TABLE II
NUMBER OF IMAGES USED FOR TRAINING AND TESTING SET IN THE HUMANeva-I DATASET.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>641</td>
<td>591</td>
</tr>
<tr>
<td>S2</td>
<td>439</td>
<td>438</td>
</tr>
<tr>
<td>S3</td>
<td>448</td>
<td>450</td>
</tr>
<tr>
<td>All</td>
<td>1528</td>
<td>1479</td>
</tr>
<tr>
<td>Jogging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>227</td>
<td>358</td>
</tr>
<tr>
<td>S2</td>
<td>389</td>
<td>398</td>
</tr>
<tr>
<td>S3</td>
<td>402</td>
<td>402</td>
</tr>
<tr>
<td>All</td>
<td>1027</td>
<td>1158</td>
</tr>
</tbody>
</table>

| All        | 2555     | 2637    |

Fig. 9. Mixing coefficient ($\alpha$) versus variance $\sigma^2$ for different values of control parameter $\lambda$

distribution from the Gaussian Process regressors are discarded. This case is equivalent to the annealed particle filter (APF). It is to be noted that all three cases use annealing for pose tracking. We also evaluated the performance of a standard particle filter (PF) and the GP-PF that does not use annealing for tracking.

1) Parameter Selection: There are four parameters that need to be set in the tracking system. The first one is the value of $\lambda$ in Eq. (19). Figure 9 shows the relationship of $\alpha$ in terms of $\sigma^2$ for different values of the control parameter $\lambda$. We empirically observed that the predictions from MGP regressor whose variance ($\sigma^2$) is greater than 0.015 have larger pose estimation errors. In order to suppress the influence of such predictions on tracking, $\lambda$ is set to a value so that $\alpha$ becomes 0 when $\sigma^2 > 0.015$. We therefore select $\lambda = 0.003$ so that $\alpha$ becomes zero for $\sigma^2 > 0.015$.

The second parameter is the sampling diagonal covariance matrix $\Omega$ of the motion model. The diagonal components of $\Omega$ correspond to the sampling variance of each body angle. They are computed so that the standard deviation of each body angle is set to equal the maximum absolute inter frame angular difference for a particular activity [27].

The third free parameter is the number of particles $n$. We found its optimal value via validation. Figure 10
Fig. 10. Number of particles \(n\) versus mean 3D pose tracking errors for different pose tracking methods.

### TABLE III

**Comparison of 3D pose estimation errors of the MGP regressor for different types of gating functions on Walking and Jogging activities.**

<table>
<thead>
<tr>
<th>Gating function type</th>
<th>3D pose estimation error in mm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3D pose estimation error in mm</td>
</tr>
<tr>
<td></td>
<td>variance</td>
</tr>
<tr>
<td>Walking</td>
<td>47.22 ± 33.08</td>
</tr>
<tr>
<td>Jogging</td>
<td>52.82 ± 30.0</td>
</tr>
</tbody>
</table>

plots the values of \(n\) versus the mean 3D errors with respect to the different tracking methods. The mean 3D errors were computed using 150 images of a walking human subject. It can be seen that for the case of AGP-PF, APF and GP-PF, the performance did not improve for \(n > 300\). Hence we set the optimal value of \(n = 300\). Our proposed AGP-PF tracking gave the lowest error for all number of particles. The figure also depicts the advantage of annealing by an improved performance of AGP-PF and APF over the performance of GP-PF and PF.

The fourth free parameter is the number of annealing layers \(M\). We set \(M = 5\), as the pose tracking performance does not improve for \(M > 5\). Table III compares the pose estimation performance of the MGP regressor for different choices of gating functions on the Walking and Jogging sequences of the dataset. The performance of MGP regressor is superior when the multinomial regressor is used. We therefore incorporate the multinomial logistic regressor in the gating function of our MGP regressor. It should be noted that in order to compute the 3D pose estimation error, the joint locations are measured relative to the torso joint.

2) **Experimental Results:** Table IV (a) and IV (b) show the mean 3D tracking errors for the three cases for subjects S1, S2 and S3 for the Walking and Jogging activities. Table IV (a) shows the tracking errors for a single camera whereas Table IV (b) shows the tracking errors for three cameras. Also included in the tables are the standard deviations of these errors. The results show that our AGP-PF with a dynamic mixing coefficient gave the lowest mean errors. It also produced the smallest standard deviations denoting that the estimated poses using our method are more stable. It can be seen that the standard particle filter with annealing produced a larger error because tracking failed at an early stage. Although the pose detection method produced smaller errors than the
TABLE IV

THE MEAN AND STANDARD DEVIATION OF 3D TRACKING ERRORS IN MM OF AGP-PF, AGP APF. THE TRACKING WAS PERFORMED ON IMAGES CAPTURED BY (A) THREE CAMERAS AND (B) A SINGLE CAMERA.

(a) Tracking from multiple cameras

<table>
<thead>
<tr>
<th>Method</th>
<th>AGP-PF (mm)</th>
<th>AGP (mm)</th>
<th>APF (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>42.4 ± 12.8</td>
<td>56.1 ± 31.1</td>
<td>142.1 ± 86.8</td>
</tr>
<tr>
<td>S2</td>
<td>34.1 ± 21.5</td>
<td>45.8 ± 20.3</td>
<td>66.9 ± 23.8</td>
</tr>
<tr>
<td>S3</td>
<td>62.9 ± 34.6</td>
<td>69.8 ± 36.5</td>
<td>145.6 ± 32.7</td>
</tr>
<tr>
<td>Jogging</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>70.9 ± 23.9</td>
<td>86.1 ± 28.6</td>
<td>169.6 ± 75.1</td>
</tr>
<tr>
<td>S2</td>
<td>50.6 ± 16.4</td>
<td>61.3 ± 20.2</td>
<td>80.8 ± 35.5</td>
</tr>
<tr>
<td>S3</td>
<td>55.1 ± 20.2</td>
<td>59.3 ± 28.4</td>
<td>144.3 ± 80.9</td>
</tr>
</tbody>
</table>

(b) Tracking from a single camera

<table>
<thead>
<tr>
<th>Method</th>
<th>AGP-PF (mm)</th>
<th>AGP (mm)</th>
<th>APF (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>110.5 ± 91.7</td>
<td>143.6 ± 122.3</td>
<td>321.6 ± 153.2</td>
</tr>
<tr>
<td>S2</td>
<td>78.6 ± 42.4</td>
<td>104.9 ± 7</td>
<td>225.2 ± 115.9</td>
</tr>
<tr>
<td>S3</td>
<td>128.1 ± 86.5</td>
<td>164.0 ± 100.6</td>
<td>403.3 ± 145.1</td>
</tr>
<tr>
<td>Jogging</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>139.9 ± 85.6</td>
<td>174.4 ± 91.6</td>
<td>303.4 ± 113.6</td>
</tr>
<tr>
<td>S2</td>
<td>124.6 ± 86</td>
<td>136.6 ± 85.4</td>
<td>177.7 ± 87.4</td>
</tr>
<tr>
<td>S3</td>
<td>144.2 ± 83</td>
<td>176.95 ± 94</td>
<td>414.3 ± 182.4</td>
</tr>
</tbody>
</table>

APF, the standard deviations of the errors were larger than those from AGP-PF. Moreover, AGP-PF with dynamic mixing coefficients was able to more accurately track the pose over the frames and recover from mistracking.

Figure 11 (a) and (b) show the pose tracking errors of the Walking sequence for all three cases for single and multiple camera tracking. It can be seen that AGP-PF gave the lowest errors for most of the frames and provided more stable pose estimates than the other two cases for both single and multiple camera tracking. AGP-PF performs tracking by giving higher weights to the correct output poses from GP regression while discarding incorrect output poses. Examples of how incorrect output poses from GP regression are given less weights for tracking are shown in Fig. 12. In this case, our AGP-PF gives more weights to the poses sampled from the motion model and hence gives correct predictions.

Figure 13 shows an example of a multi-modal pose output given by the mixture of GP models. The 3D pose estimates of the two GP regressors with the highest gating probability are shown in Fig. 13 (c) and (d). These two probable solutions denote the pose ambiguities associated with the silhouette. Our AGP-PF tracking method uses them to predict the correct 3D pose as shown in Fig. 13 (e). Figure 14 compares the knee flexion angles estimated using our method with the ground truth knee flexion angles for each video frame. Figures 15 and 16 display some of the output poses predicted using our method for the Walking and Jogging sequences. The faster and larger arm and leg movements of the human subject in the Jogging sequence makes the pose estimation problem more challenging. Our experiments show that our proposed AGP-PF can effectively track 3D pose in a video sequence.
Fig. 11. Comparison of the 3D tracking errors in mm for the Walking sequence evaluated from AGP-PF, APF and AGP pose tracking methods for (a) single camera and (b) multiple cameras.

Fig. 12. Examples of the 3D poses obtained from the GP regression (rendered in blue color) and AGP-PF tracking (rendered in green color). Below each output is the corresponding value of $\alpha$. Each 3D pose is illustrated using the boundaries of the projected cylinders of the 3D pose (Figure best viewed in color).

Examples of videos to illustrate the tracking of the 3D pose are available.¹

¹http://www.csse.uwa.edu.au/~suman/videos

Fig. 13. (a) An image from the test set. (b) Corresponding silhouette image. (c) & (d) The 3D pose estimates of two GP regressors with the highest gating probability are displayed, with red denoting left limbs and blue denoting right limbs. The associated gating probabilities are displayed below each output pose. These two probable solutions denote the pose ambiguities associated with the silhouette. (e) Our AGP-PF used them as input for 3D pose tracking (Figure best viewed in color).
Fig. 14. Comparison of the ground truth and the estimated knee flexion angle using our proposed AGP-PF method.

![Graph showing comparison of ground truth and estimated angles.](image1)

Fig. 15. Pose estimation results on some of the images of the test set of Walking sequence. Each 3D pose is illustrated using the boundaries of the projected cylinders of the 3D pose (Figure best viewed in color).

![Pose estimation results on images.](image2)

D. Comparison with other works

We compared our work with the state of the art tracking methods [4], [32], [34], [58]. These approaches use annealed particle filter [58], smoothing particle filter [63], particle filter with physics based prior [34], local optimization [4], a hybrid of local and global optimization [32] and a latent variable model [43] for 3D pose tracking.

Table V shows our pose tracking results compared to a smoothing particle filter of Ref. [63] and a latent variable model of Ref. [43] on the HumanEva-I dataset. The results show that our AGP-PF outperforms the method of Ref. [63] for all the subjects. We found that the mean 3D tracking error of our approach is lower than the error of Ref. [43] for Subjects S1 and S2. The mean 3D tracking error of our approach is higher than the error of Ref. [43] for subject S3.

Table VI shows our pose tracking results compared to those from other state of the art approaches for the Walking
Fig. 16. Pose estimation results on some of the images of the test set of Jogging sequence. Each 3D pose is illustrated using the boundaries of the projected cylinders of the 3D pose. In comparison with the Walking sequence in Fig. 15, because of the larger and faster arm and leg movements, the estimated poses for this sequence are less accurate (Figure best viewed in color).

<table>
<thead>
<tr>
<th>Table V</th>
<th>Comparison of mean 3D tracking errors (mm) of our AGP-PF method with other approaches on the Walking activity of the HumanEva-I dataset. The standard deviation over frames is not provided by [43].</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref/Subject</td>
<td>3D tracking error in mm</td>
</tr>
<tr>
<td>Peursun [63]</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td>S2</td>
</tr>
<tr>
<td></td>
<td>S3</td>
</tr>
<tr>
<td>Yao [43]</td>
<td>44.0</td>
</tr>
<tr>
<td></td>
<td>45.5</td>
</tr>
<tr>
<td><strong>Our method</strong></td>
<td><strong>42.4 ± 12.8</strong></td>
</tr>
</tbody>
</table>

sequence of Subject S2 of the HumanEva-II dataset. The camera view of the dataset which is used to train the discriminative model is different from the view of Camera C1 of HumanEva-II dataset which is used as the test sequence. Therefore we normalize the view (align with respect to the test camera view) of the pose predicted by the discriminative model before tracking. With this pre-processing step, we obtained a mean tracking error of 73.0 mm which surpasses the performance of Refs. [58] and [4]. This shows that our method is not only able to generalize well w.r.t the subjects’ gaits and genders, but it is also able to generalize well between camera views.

We also compared our algorithm to the approaches in Refs. [34] and [32] and it turned out that they achieve a better performance (an average error of 37 mm and 53 mm respectively). This is understandable because Ref. [34] uses strong priors based on a complex bio-mechanical model of the human body and Ref. [32] uses a local optimization with a surface-based 3D human model whereby a person’s body is scanned off-line (using an expensive body scanner). This phase requires the full cooperation of the person which is not required in our case.
TABLE VI
3D pose tracking errors of our AGP-PF method and some state of art pose tracking approaches on the HumanEva-II dataset. The walking activity of subject S2 is used in the experiment.

<table>
<thead>
<tr>
<th>Ref</th>
<th>3D tracking error in mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brubaker [34]</td>
<td>53 ± 9</td>
</tr>
<tr>
<td>Gall [32]</td>
<td>37.5</td>
</tr>
<tr>
<td>Sigal [58]</td>
<td>76 ± 5</td>
</tr>
<tr>
<td>Corazza [4]</td>
<td>78 ± 10</td>
</tr>
<tr>
<td>Our method</td>
<td>73.0 ± 6.5</td>
</tr>
</tbody>
</table>

VI. DISCUSSION AND CONCLUSIONS

We have presented an Annealed Gaussian Process Guided Particle Filter (AGP-PF) for 3D human pose tracking in video sequences. Our method effectively exploits the discriminative distribution obtained from the mixture of Gaussian Process regression model with a motion model to obtain an accurate and stable tracking of the 3D pose in each video frame. We use the prediction uncertainty obtained from the Gaussian Process regression to dynamically determine the contribution of the discriminative model for tracking. Our method does not require initialization and can resolve pose ambiguities during tracking using motion information and multi-view images. Experimental results show that our proposed AGP-PF can accurately track the 3D pose in a long video sequence. Although this paper uses a stationary motion model for pose tracking, we believe that by using a learned motion model, the pose can be tracked more accurately. Tracking in a lower dimensional latent space could also be employed for a more efficient performance. Moreover, the scalability of our method could be improved by training the discriminative model using data from various other activities.

ACKNOWLEDGMENT

This work was in part supported by the ARC Discovery Project grant DP0771294.

REFERENCES


Suman Sedai received his MSc from Inha University, South Korea and his PhD from The University of Western Australia in 2007 and 2012 respectively. His research interests include image processing, visual tracking, object recognition, pattern recognition and machine learning. He is currently a postdoctoral researcher at IBM Research – Australia.
Mohammed Bennamoun received his M.Sc. from Queen’s University, Kingston, Canada in the area of Control Theory, and his PhD from Queen’s /Q.U.T in Brisbane, Australia in the area of Computer Vision. He lectured Robotics at Queen’s, and then joined QUT in 1993 as an Associate Lecturer. He is currently a Winthrop Professor. He served as the Head of the School of Computer Science and Software Engineering at The University of Western Australia (UWA) for five years (February 2007-March 2012). He served as the Director of a University Centre at QUT: The Space Centre for Satellite Navigation from 1998-2002. He was an Erasmus Mundus Scholar and Visiting Professor in 2006 at the University of Edinburgh. He was also Visiting Professor at CNRS (Centre National de la Recherche Scientifique) and Telecom Lille1, France in 2009, The Helsinki University of Technology in 2006, and The University of Bourgogne and Paris 13 in France in 2002-2003. He is the co-author of the book “Object Recognition: Fundamentals and Case Studies”, Springer-Verlag, 2001, and the co-author of an Edited book on “Ontology Learning and Knowledge Discovery Using the Web”, published in 2011. He published over 250 journal and conference publications, and secured highly competitive national grants from the Australian Research Council (ARC). Some of these grants were in collaboration with Industry partners (through the ARC Linkage Project scheme) to solve real research problems for industry, including Swimming Australia, the West Australian Institute of Sport, a textile company (Beaulieu Pacific), and AAM-GeoScan. He worked on research problems and collaborated (through joint publications, grants and supervision of PhD students) with researchers from different disciplines including Animal Biology, Speech Processing, Biomechanics, Ophthalmology, Dentistry, Linguistics, Robotics, Photogrammetry, and Radiology. He won the Best Supervisor of the Year Award at QUT. He also received an award for research supervision at UWA in 2008. He served as a guest editor for a couple of special issues in International journals, such as the International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI). He was selected to give conference tutorials at the European Conference on Computer Vision (ECCV) and the International Conference on Acoustics Speech and Signal Processing (IEEE ICASSP). He organized several special sessions for conferences; including a special session for the IEEE International Conference in Image Processing (IEEE ICIP). He was on the program committee of many international conferences. He also contributed in the organisation of many local and international conferences. His areas of interest include control theory, robotics, obstacle avoidance, object recognition, artificial neural networks, signal/image processing and computer vision (particularly 3D).

Du Huynh is an Associate Professor at the School of Computer Science and Software Engineering, The University of Western Australia. She obtained her Ph.D in Computer Vision, in 1994, at the same university. Since then, she has worked for the Australian Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP) and Murdoch University. She has been a visiting scholar at Lund University, Malmö University, Chinese University of Hong Kong, Nagoya University, Gunma University, and the University of Melbourne. Associate Professor Huynh has won several grants funded by the Australian Research Council. Her research interests include shape from motion, multiple view geometry, video image processing, and visual tracking.